Bacharel em Engenharia de Software

Árvore Binária: AVL

Estrutura de Dados II

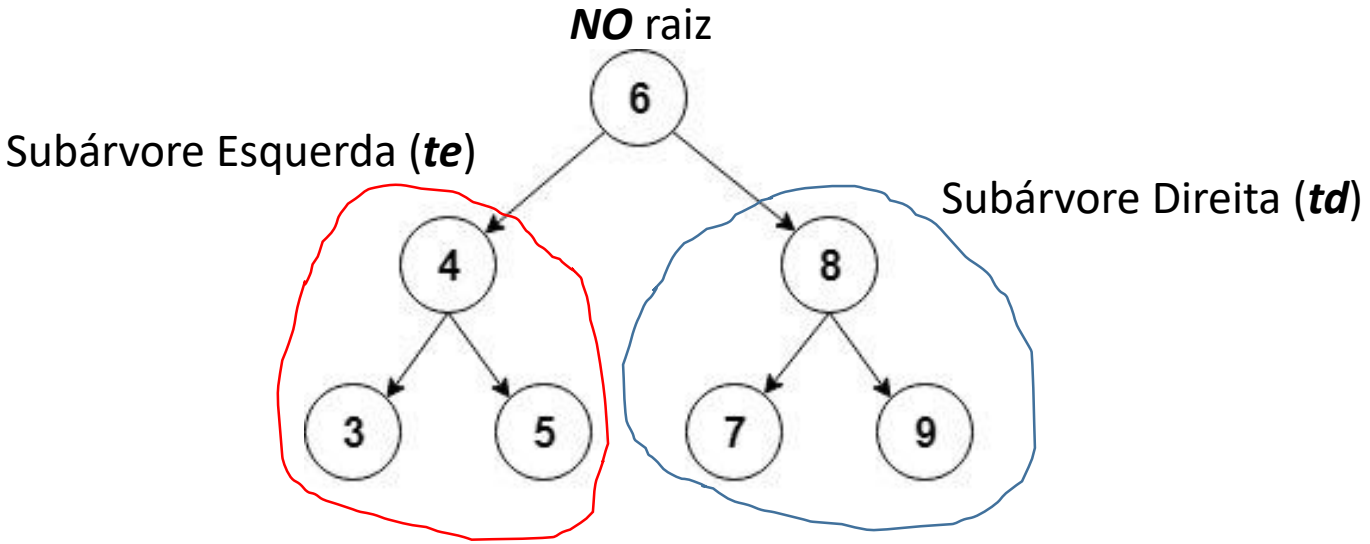
INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
AMAZONAS

www.ifam.edu.br

Benevaldo Gonçalves
benevaldo.goncalves@ifam.edu.br
(92) 99301 - 0683

Árvore Binária: Conceito

- Árvore Binária **T** (tree) é um conjunto finito de elementos denominados ***nós*** ou ***vértices***, talque:
  - **T** = 0, árvore é vazia ou
  - Existe um ***NÓ r***, chamado ***raiz de T***, e os ***NÓS*** restantes podem ser divididos em dois subconjuntos disjuntos, ***te*** (árvore esquerda) e ***td*** (árvore direita), que são subárvores as quais, por sua vez, também são árvores binárias.
  - Cada NÓ deve ter no máximo DOIS filhos;
  - Um NÓ que não possua filhos chama-se Folha.



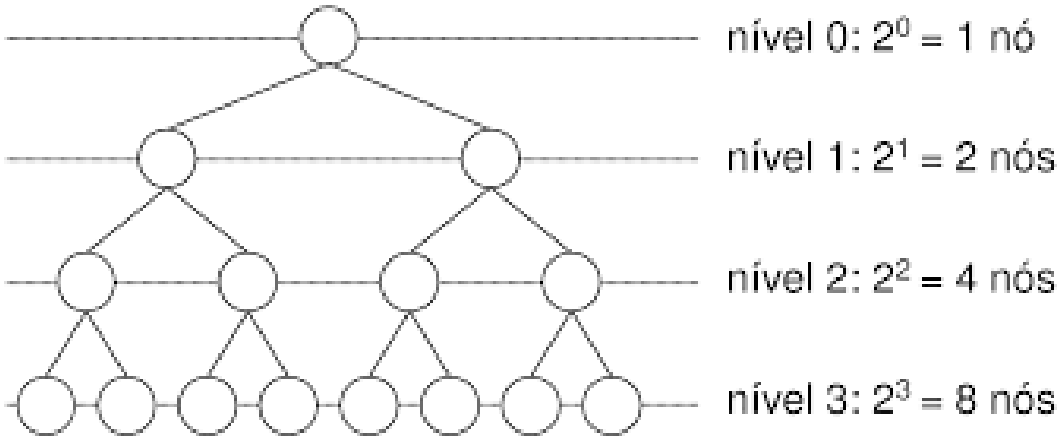
NÓ Folhas: 3, 5, 7, e 9

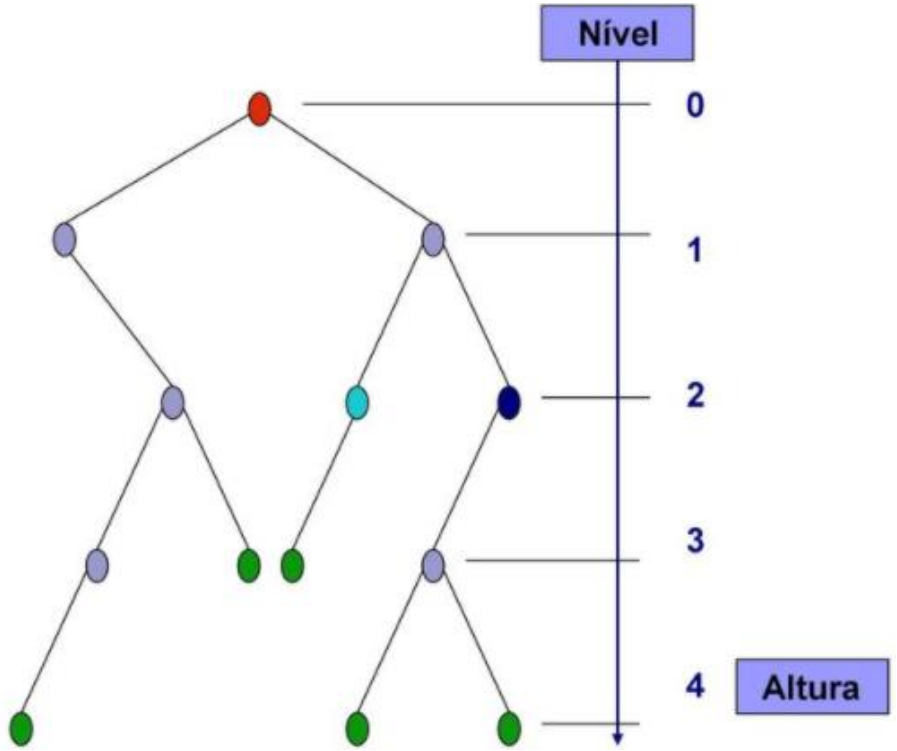
Árvore Binária: Propriedades

- Filhos da esquerda têm valores ***menores*** do que a raiz;
- Filhos da direita têm valores ***maiores ou iguais*** do que a raiz;
- Para cada NÓ, o menor valor fica a esquerda e o maior ou igual fica a direita;
- O ***NÍVEL*** de um NÓ ***n*** pode ser definido da seguinte forma: o NÓ raiz tem nível ***0***, os outros NÓS tem ***um nível*** que é ***maior uma unidade*** que o nível de seu respectivo NÓ ***pai***;
- A ***altura*** de uma árvore binária é igual ao ***maior nível***, a partir do ***NÓ raiz***, da árvore binária.
- A ***altura*** de cada ***NÓ***, corresponde ao seu maior nível, a partir do NÓ analisado.
- Nó do tipo ***Folhas*** (nó analisado) não possuem ***níveis*** e consequentemente não possuem **altura**.

Árvore Binária: Propriedades

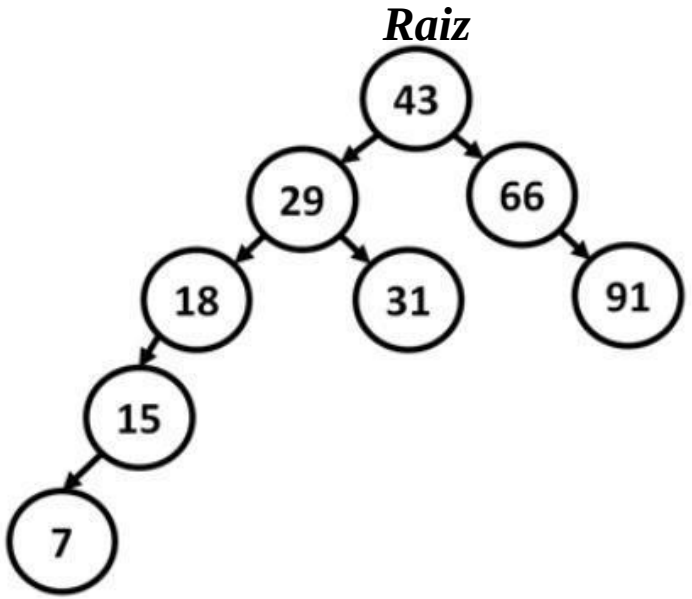
- O número do NÍVEL determina o número máximo de NÓS.
- A profundidade de um nó é a distância deste nó até a raiz. Um conjunto de nós com a mesma profundidade é denominado nível da árvore. A maior profundidade de um nó, é a altura da árvore.





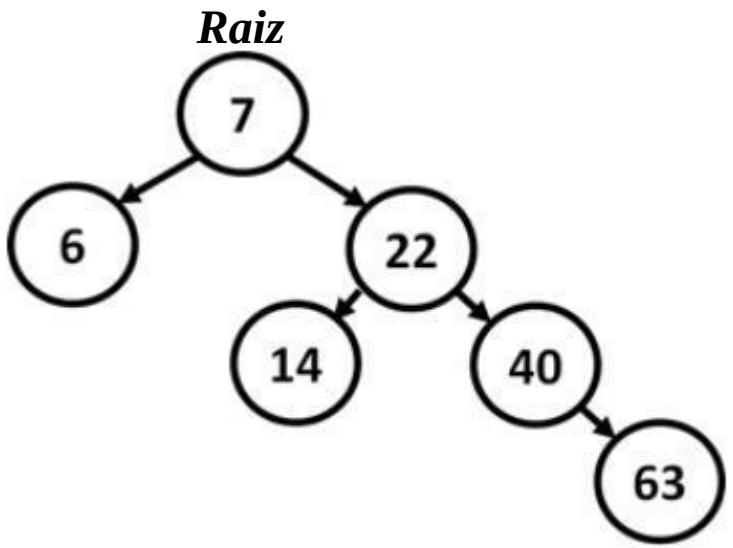
Árvore Binária: Problema

- Uma estrutura de dados do tipo árvore binária, devido as suas características torna-se uma eficiente estrutura para armazenar e pesquisar dados, porém existem possibilidades na sua geração que pode causar grandes problemas no seu processo de pesquisa. As árvores binárias, a baixo, são um exemplo:



Árvore binária desbalanceada a esquerda

Em ambas as árvores o processo de busca não seria eficiente para um cenário ideal, pois haveria um custo maior de processamento para realizar uma busca tanto do lado esquerdo quando do lado direito.

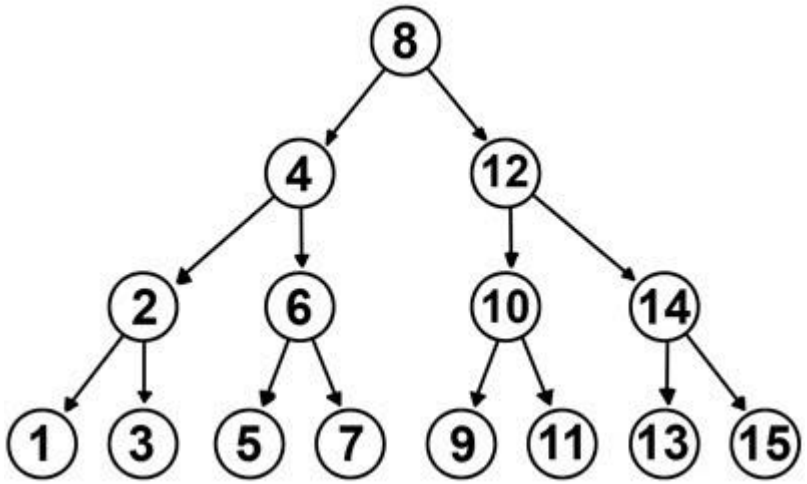


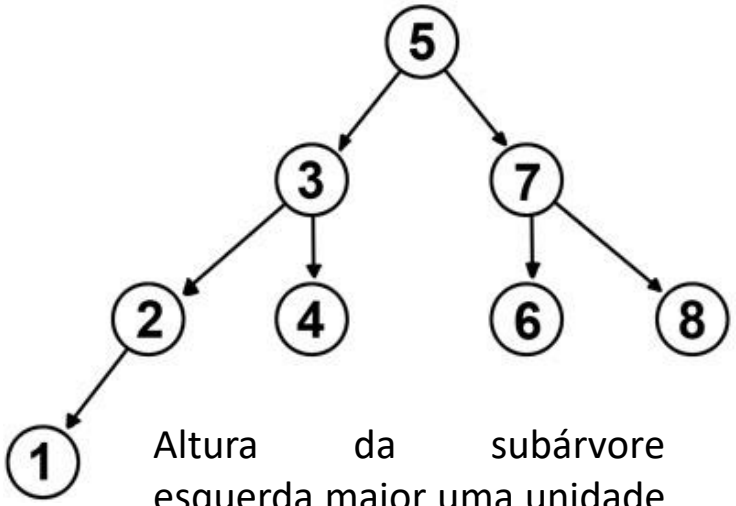
Árvore binária desbalanceada a direita

Árvore Binária: Solução (AVL)

- A AVL (**A**delson-**V**elskii e **L**andis – 1962) é uma árvore binária ***altamente balanceada***, isto é, nas ***inserções*** e ***exclusões***, procura-se executar uma rotina de ***balanceamento*** tal que as ***alturas*** das sub-árvores esquerda e sub-árvores direita tenham alturas bem próximas;
- Uma árvore AVL é uma árvore na qual as alturas das subárvores esquerda e direita de cada nó ***diferem no máximo por uma unidade***.

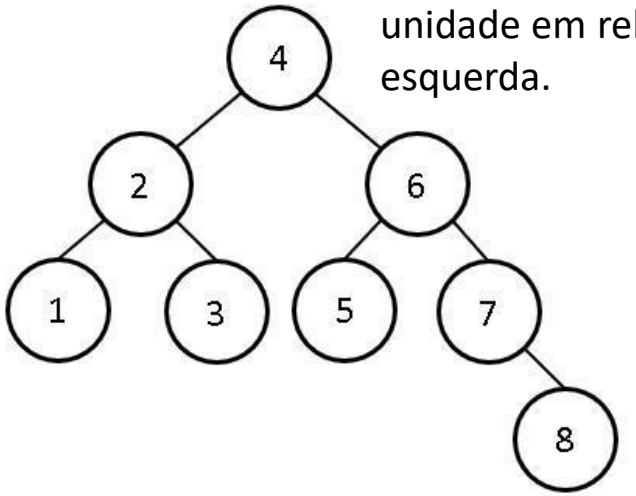
Altura da subárvore esquerda igual ao da direita.





Altura da subárvore esquerda maior uma unidade em relação a direita.

Altura da subárvore direita maior uma unidade em relação a esquerda.

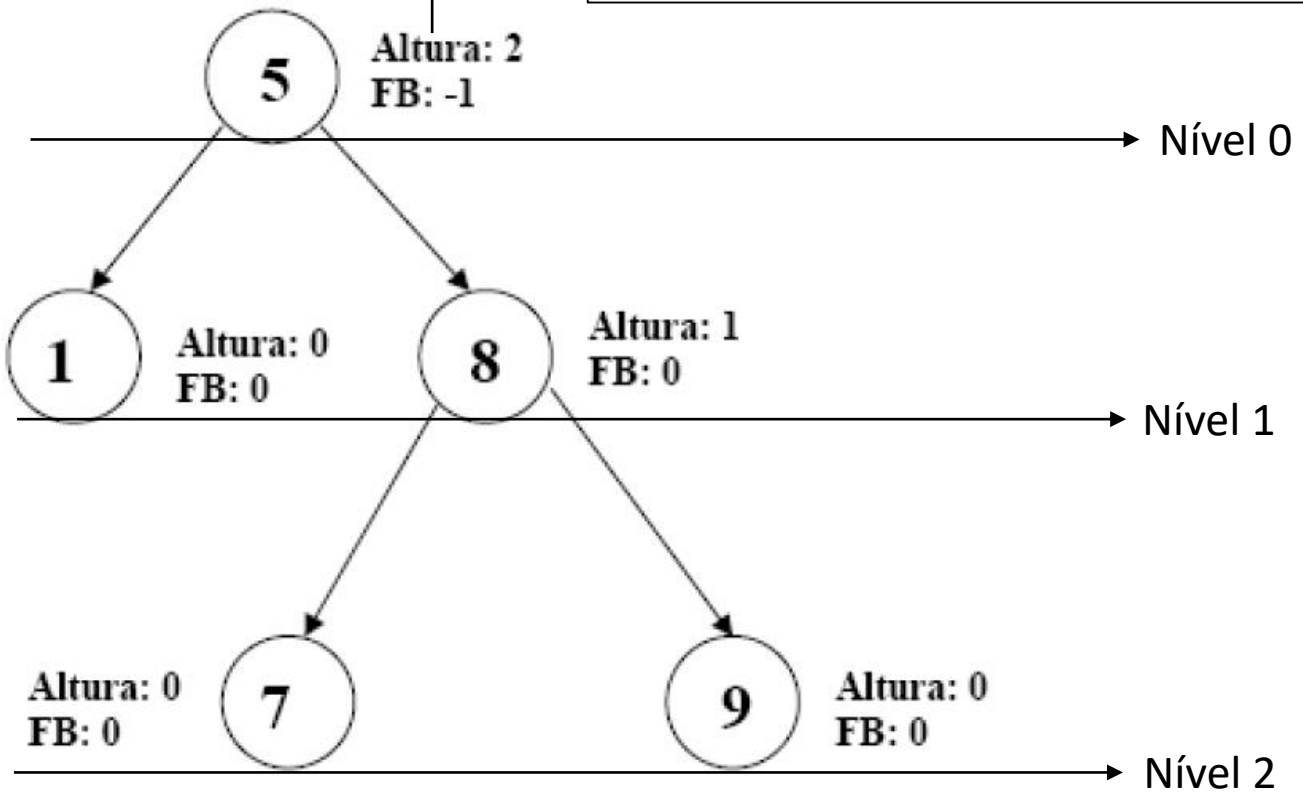


Árvore Binária AVL
Fator de Balanceamento

- O balanceamento de um ***NÓ*** é definido como a **altura** de sua sub-árvore ESQUERDA ***menos*** a **altura** de sua sub-árvore DIREITA. Esta definição chama-se ***FATOR DE BANLANCEAMENTO*** (FB).
- Cada NÓ numa árvore binária balanceada (AVL) tem balanceamento de 1, -1 ou 0.
- Se o valor do balanceamento do NÓ for **diferente** de 1, -1 ou 0. Essa árvore **NÃO** é balanceada.
- Se a probabilidade de pesquisar um dado for a mesma para todos os dados, uma árvore binária balanceada determinará a busca mais eficiente.

Árvore Binária AVL
Fator de Balanceamento (FB)

Altura da árvore binária, maior nível = 2
Altura da sub-árvore Esquerda = 1
Altura da sub-árvore Direita = 2
Fator de Balanceamento (***FB***) = 1 – 2 = -1



Realiza-se a mesma análise para todos os NÓS com o objetivo de encontrar o Fator de Balanceamento (FB) de cada NÓ, após este procedimento verificar SE TODOS os NÓS estão com o FB igual a 1, -1 ou 0. Caso afirmativo a árvore binária está balanceado, portanto trata-se de uma AVL.

FB = altura da sub-árvore Esquerda – altura da Direita

FB = 0, indica altura da sub-árvore esquerda = direita

FB = 1, indica altura da sub-árvore esquerda > direita

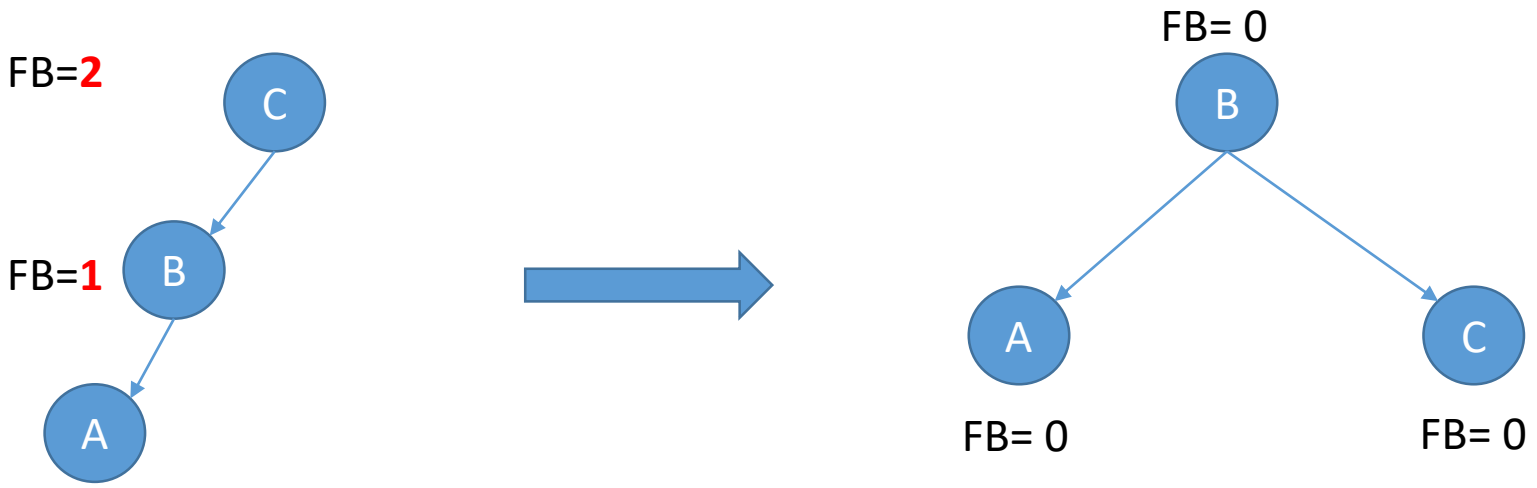
FB = -1, indica altura da sub-árvore esquerda < direita

AVL: Algoritmo Rotação

- A operação básica em uma árvore AVL geralmente envolve os mesmos algoritmos de uma árvore de busca binária desbalanceada. A rotação na árvore AVL ocorre devido ao seu ***desbalanceamento***.
- Uma ***rotação simples*** ocorre quando um ***nó*** está desbalanceado e seu ***filho*** estiver no ***mesmo sentido da inclinação***, formando uma linha reta.
- Uma ***rotação-dupla*** ocorre quando um ***nó*** estiver desbalanceado e seu ***filho*** estiver inclinado no ***sentido inverso ao pai***, formando um "joelho".
- Para garantirmos as propriedades da árvore AVL, rotações devem ser feitas conforme necessário após operações de ***remoção*** ou ***inserção***. Seja ***P*** o nó pai, ***Fe*** o filho da esquerda de ***P*** e ***Fd*** o filho da direita de ***P*** podemos definir 4 tipos diferentes de rotação: ***rotação simples a direita, rotação simples a esquerda, rotação dupla à direita e rotação dupla à esquerda***.

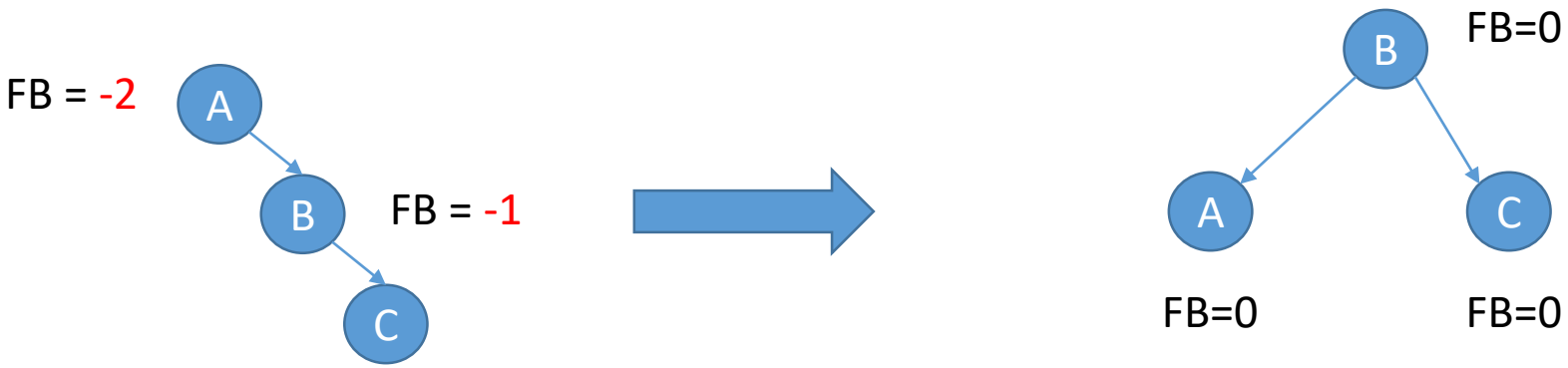
AVL: Rotação simples à direita

- Deve ser efetuada quando a diferença das alturas ***h*** dos filhos de ***P*** é igual a **2** e a diferença das alturas ***h*** dos filhos de ***Fe*** (filho da esquerda de P) é igual a **1**. O nó ***Fe*** deve tornar o novo pai e o nó P deve se tornar o filho da direita de ***Fe***.



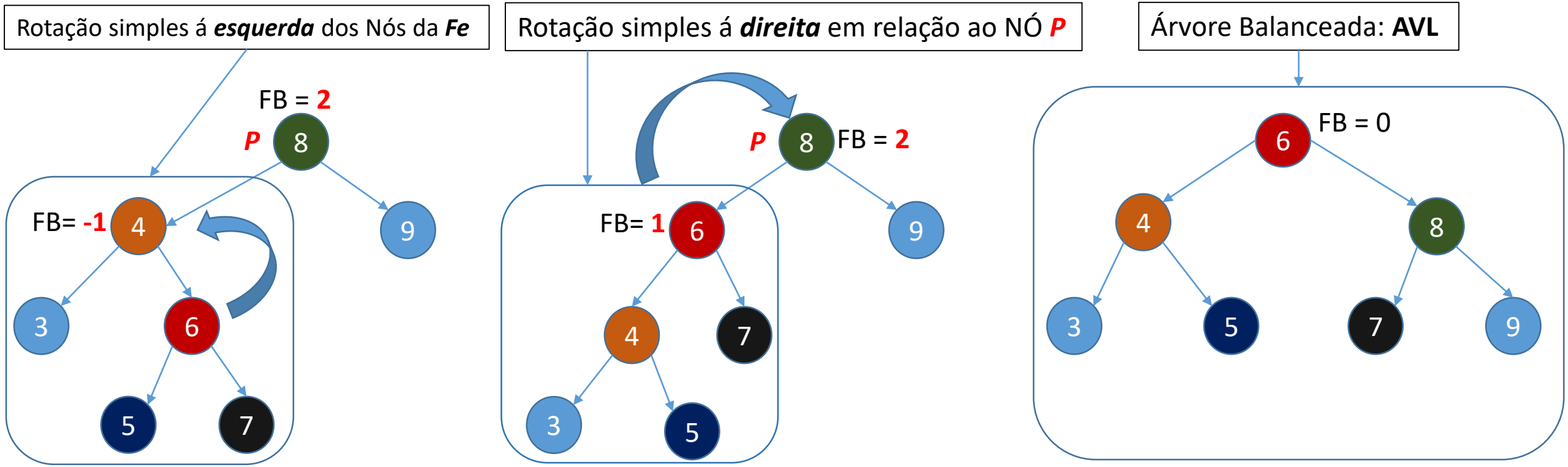
AVL: Rotação simples à esquerda

- Deve ser efetuada quando a diferença das alturas ***h*** dos filhos de ***P*** é igual a ***-2*** e a diferença das alturas ***h*** dos filhos de ***Fd*** (filhos da direita de P) é igual a ***-1***. O nó ***Fd*** deve tornar o novo pai e o nó ***P*** deve se tornar o filho da esquerda de ***Fd***.



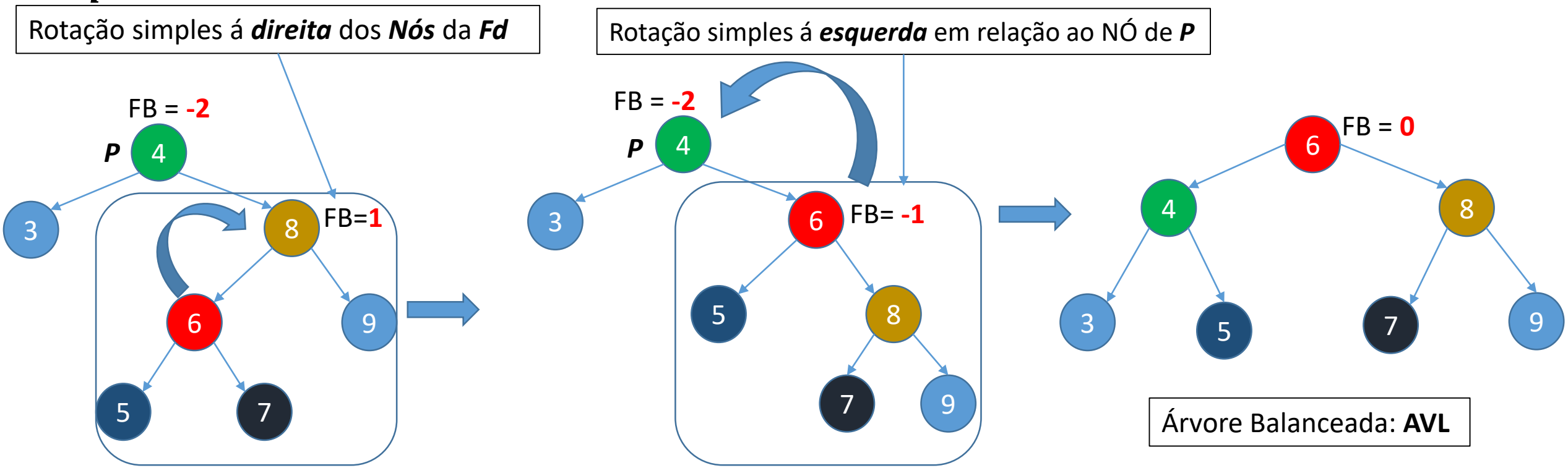
AVL: Rotação dupla à direita

- Deve ser efetuada quando a diferença das alturas ***h*** dos filhos de ***P*** é igual a ***2*** e a diferença das alturas ***h*** dos filhos de ***Fe*** (filhos da subárvore esquerda de ***P***) é igual a ***-1***. Nesse caso devemos aplicar uma rotação à ***esquerda*** no nó ***Fe*** e, em seguida, uma rotação à ***direita*** no nó ***P***.



AVL: Rotação dupla à esquerda

- Deve ser efetuada quando a diferença das alturas ***h (FB)*** dos filhos de ***P*** é igual a ***-2*** e a diferença das alturas ***h (FB)*** dos filhos de ***Fd*** (filhos a direita de P) é igual a ***1***. Nesse caso devemos aplicar uma rotação à ***direita*** no nó ***Fd*** e, em seguida, uma rotação à ***esquerda*** no nó ***P***.



AVL: Algoritmo

- Por definição, todos os nós da AVL devem ter fator de balanceamento (fb = -1, 0 ou 1). Para garantir essa propriedade, a cada ***inserção*** ou ***remoção*** o fator de balanço deve ser atualizado a partir do ***pai do nó inserido até a raiz da árvore***. Na inserção basta encontrar o primeiro nó ***desregulado (fb= -2 ou fb= 2)***, aplicar o operação de ***rotação necessária***, não havendo necessidade de verificar os demais nós. Na ***remoção*** a verificação deverá prosseguir até a raiz, podendo requerer mais de uma ***rotação***.
- A busca é a mesma utilizada em árvore binária de busca. A busca pela chave de valor K inicia sempre pelo nó raiz da árvore.

AVL: Algoritmo de Inserção

- Para inserir um novo nó de valor **K** em uma **árvore AVL** é necessária uma busca por **K** nesta mesma árvore. Após a busca o local correto para a inserção do nó **K** será em uma subárvore vazia de uma folha da árvore. Depois de inserido o nó, a ***altura*** do nó ***pai*** e de ***todos os nós acima deve ser atualizada***. Em seguida o algoritmo de ***rotação simples*** ou ***dupla*** deve ser acionado para o primeiro nó pai ***desregulado (fator de balanceamento ≠ -1, 1, ou 0)***.

Como identificar mudança de altura na Inserção?

- Considerar ***p*** (NÓ), ***Tp*** (subárvore), ***fb*** (fator de balanceamento).
- Considerar que o nó ***p*** é ***raiz*** da subárvore ***Tp*** e houve inserção em uma de suas subárvores.
- Caso a subárvore Tp tenha mudado de altura, ***decrementar fb*** (inserção na subárvore ***esquerda***) ou ***incrementar fb*** (inserção na subárvore direita).
- Caso 1: Ao inserir um nó folha, a subárvore Tp passa de altura 0 para altura 1, então Tp mudou de altura.
- Caso 2: fb = 0 ***antes da inserção*** foi alterado para 1 ou -1, então a subárvore Tp mudou de altura.
- Caso 3: fb = 1 ou -1 ***antes da inserção***, passou a ter valor 0, então a subárvore Tp não mudou de altura.
- ***Caso 4***: O fb passou a ter valor -2 ou 2 ***após a inserção***, então há necessidade de aplicação de alguma ***operação de rotação***. Após a rotação, a subárvore Tp terá a mesma altura ***anterior à inserção***.

AVL: Algoritmo de Remoção

- O primeiro passo para remover uma chave K consiste em realizar uma busca binária a partir do ***nó raiz***. Caso a busca encerre em uma subárvore vazia, então a chave K não está na árvore e a remoção não pode ser realizada. Caso a busca encerre em um nó ***"n"*** o nó que contenha a chave então a remoção poderá ser realizada da seguinte forma:
  - Caso 1: O nó ***n*** é uma folha da árvore, apenas exclui-lo.
  - Caso 2: O nó ***n*** tem apenas uma subárvore, necessariamente composta de um nó folha, basta apontar o nó pai de ***n*** para a única subárvore e excluir o nó ***n***.
  - Caso 3: O nó ***n*** tem duas subárvores: localizar o nó ***"p"*** predecessor ou sucessor de K, que sempre será um nó ***folha*** ou possuirá apenas uma ***subárvore***; copiar a chave de ***p*** para o nó ***n***; excluir o nó ***p*** a partir da respectiva subárvore de ***n***.
- O último passo consiste em verificar se houve desbalanceamento em algum NÓ a partir do pai do nó ***excluído*** até o ***nó raiz da árvore***. Aplicar rotação simples ou dupla em cada nó desbalanceado.

Como identificar mudança de altura na Remoção?

- Considerar que o nó ***p*** é raiz da subárvore T_p e houve remoção em uma de suas subárvores.
- Caso a subárvore T_p tenha mudado de altura, incrementar fb (remoção na subárvore esquerda) ou decrementar fb (remoção na subárvore direita).
- Caso 1: Ao remover um nó folha, a subárvore T_p passa de altura 1 para altura 0, então T_p mudou de altura.
- Caso 2: fb = 0 antes da remoção foi alterado para 1 (remoção à esquerda) ou -1 (remoção à direita), então a subárvore T_p não mudou de altura.
- Caso 3: fb = 1 ou -1 antes da remoção à direita e à esquerda, respectivamente, passando a ter valor 0, então a subárvore T_p mudou de altura.
- ***Caso 4***: fb passou a ter valor -2 ou 2 após a remoção, então houve necessidade de aplicação de rotação. Após a rotação dupla T_p muda de altura.